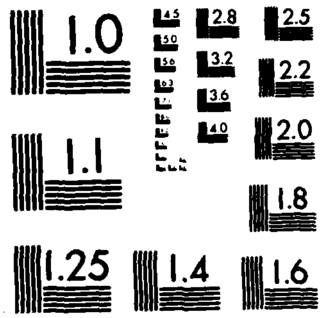
AD-A139 972   A NEW DISTRIBUTED ROUTING PROTOCOL(U) ARMY COMMUNICATIONS-ELECTRONICS COMMAND FORT MONMOUTH NJ R J KIM ET AL. JAN 84 CECOM-TR-84-1   1/1

UNCLASSIFIED   F/G 9/4   NL

END
DATE
FILMED
5 84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

RESEARCH AND DEVELOPMENT TECHNICAL REPORT

CECOM-TR-84-1

A NEW DISTRIBUTED ROUTING PROTOCOL

RICHARD J. KIM
ISRAEL MAYK
CENTER FOR SYSTEMS ENGINEERING & INTEGRATION

JANUARY 1984

DISTRIBUTION STATEMENT

Approved for public release;
distribution is unlimited.

CECOM
U S ARMY COMMUNICATIONS-ELECTRONICS COMMAND
FORT MONMOUTH, NEW JERSEY 07703

NOTICES

Disclaimers

The citation of trade names and names of manufacturers in this report is not to be construed as official Government indorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CECOM-TR-84-1	2. GOVT ACCESSION NO. AD-A139972	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A NEW DISTRIBUTED ROUTING PROTOCOL		5. TYPE OF REPORT & PERIOD COVERED Technical Report June 1983 - Jan 1984
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard J. Kim Israel Mayk		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Communications-Electronics Command Center for Systems Engineering & Integration ATTN: DRSEL-SEI-F, Fort Monmouth, NJ 07703		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Same as above		12. REPORT DATE January 1984
		13. NUMBER OF PAGES 52
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Distributed Routing, Network, Connectivity

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Research was conducted in the area of distributed routing of messages in a digital broadcast network.

The early detection routing protocol (EDRP) was developed to achieve automated shortest-path routing and optimum resource distribution. EDRP can be implemented either on a real time basis or as a network simulation tool for the network manager.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

EDRP works as follows: A minimum spanning tree is grown from a message origination point (source) until the message destination point (sink) is found. The path between the source and the sink within the minimum spanning tree is the shortest path between the two points.

A graphics simulation program was developed to test the protocol. For each tested network, minimum-hop path routing and congestion control were demonstrated.

CONTENTS

FIGURES

TABLES

A NEW DISTRIBUTED ROUTING PROTOCOL

1.0 INTRODUCTION

Research is being conducted at the Center for Systems Engineering and Integration of CECOM, Fort Monmouth, NJ, to develop a distributed routing and resource allocation protocol for use in digital radio broadcast networks. Although the original motivation for the research was to develop a new distributed routing protocol for real time use to achieve automated shortest-path routing and optimum resource distribution in a battlefield data distribution system, the research effort has evolved into developing effective computer assisted capabilities and techniques for use by the network manager. The present goal of our ongoing research is to explore the possibility of meeting the baseline for the computer assisted capabilities and techniques referred to in the concluding section of the Joint Tactical Information Distribution System Planning Guide [1].

This chapter introduces the reader to the two basic battlefield data distribution system concepts: communication needlines and network connectivity. The parameters and the established protocols that we will be dealing with are those of a typical battlefield data distribution system considered for future deployment.

1.1 Needlines

A needline is a directed (one way) virtual link required between a message origination point (source) and a message destination point (sink). Computing the maximum number of needlines possible in a network is equivalent to computing the maximum number of directed virtual links possible in a network of nodes.

A network can be represented by an undirected graph G(V,E) where V is a set of vertices (or nodes) and E is a set of edges (or undirected links). If there are two vertices in V, an edge is possible in E, and if there are three vertices in V, it is possible to have up to 3 edges in E. The maximum number of possible edges in E given V is equivalent to the total number of pairwise combinations of the vertices in V. Therefore,

$$\text{the maximum number of possible edges} = \binom{n}{2} = n!/(n-2)!2! = n(n-1)/2.$$

(An edge represents an undirected link, or equivalently, two directed links directed in opposite directions.) We can conclude by substitution that there are n(n-1) needlines possible in any given network requiring needlines.

Number of Nodes	Maximum Number of Needlines
1	0
10	90
100	9900
1000	999000

Table 1-1. Nodes vs. Needlines

The maximum number of possible needlines for a given network can be tabulated. Table 1-1 shows a quadratic increase in the maximum number of possible needlines with respect to a linear increase in the number of nodes present.

In typical military applications we find an average of 3 to 5 needlines associated with a node. This is due to the hierarchical architecture of command and control where typically 3 to 5 subordinate units actively comunicate with a command and control center.

1.2 Connectivity

Under the ideal conditions of propagation in free space and no physical obstructions, line-of-sight operations are possible between all pairs of nodes. However, when nodes are tactically deployed in an environment with irregular terrain and dense foliage, large signal attenuation (typically > 140dB @25Km) and complete outages (loss of line-of-sight) can be expected.

Relays are required to route messages when a direct line-of-sight operation is unrealizable. A node can serve as a relay if there exists a line-of-sight to the node. However, there are line-of-sight connections which have high message error rates due to environmental conditions. There are also line-of-sight connections to nodes possessing high message delays. As we can see, all line-of-sight connections cannot be considered to be equal in deciding the best path to satisfy a needline, and the variations in the parameters must be taken into account.

Network connectivity influences the network capacity requirement, the path that is taken to satisfy a needline, and the likelihood of a needline being satisfied. Therefore, network connectivity is an important parameter that needs to be quantified.

Silvester and Kleinrock [9] define the degree of a node as the number of communicants of the node. But they also define a node as a communicant of itself. We will simply define the degree of a node as the number of communicants of the node, not counting itself as its communicant. The

degree of a node is defined as the total number of edges (or arcs) which couple the node to the network.

Representing a network by the undirected graph G(V,E), the average degree for a network is the number of edges in E divided by the number of vertices in V, or average degree = E[edges/vertex]. The average degree gives an indication of the connectivity of the network.

A radio network operating in a heavily vegetated environment is likely to be sparsely connected, and the sparse connectivity may be reflected by a low average degree. There are some very important problems encountered in such an environment. The next two sections investigate the problems associated with sparsely connected networks.

1.2.1 A Disconnected Network

Figure 1-1. Disconnected Network

A network in which every node is connected to at least one other node is not necessarily fully connected. In Fig. 1-1 we see a disconnected network of six nodes, each node possessing a degree of 2. Communications between the nodes in the group [A,B,C] and the nodes in the group [D,E,F] are not feasible. In general, given a fixed average degree, the probability of a disconnected network occurring increases with the increase in the number of nodes in the network.

1.2.2 Sparse Connectivity

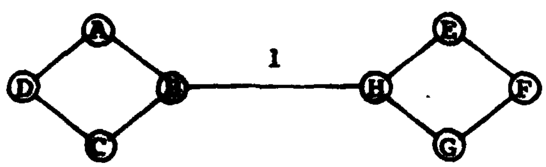


Figure 1-2. Sparsely Connected Network

A network operating in an environment of varying terrain conditions may have local variations in its network connectivity. The nodes operating in areas of sparse connectivity may encounter problems of bottlenecking.

The network in Fig. 1-2 is fully connected because each node can have a path to any other node. However, all communications between nodes A,B,C,and D and nodes E,F,G,and H must occur through link 1 which connects nodes B and H. Link 1 is where bottlenecking is likely to occur. A possible solution to the problem is to provide a line-of-sight operation between nodes A and E, and between nodes C and G.

1.2.3 Connectivity Quality

The connectivity quality (CQ) of a link is represented by an M-ary CQ symbol which requires a field of log2(M) bits. To enable the exchange of a large number of CQ's with minimum overhead capacity, M must be a small number. Letting M=4, only 2 bits per directed link are required, for which we assume the following qualification:

Ø = very high probability of transmission error
1 = high probability of transmission error
2 = medium probability of transmission error
3 = low probability of transmission error.

Figure 1-3. Route Between Nodes A and D

Suppose there is a need to transmit one hundred message blocks from node A to node D. The route chosen to satisfy the need is illustrated in Fig. 1-3. The connectivity qualities of interest are

CQ(A,B) = 3   CQ(B,C) = 2   CQ(C,D) = 1

for which we assume the following quantification:

CQ(i,j) = 3 implies Pe < 10%
CQ(i,j) = 2 implies Pe < 20%
CQ(i,j) = 1 implies Pe < 40%
CQ(i,j) = Ø implies Pe < 60%
The absence of CQ(i,j) in a message implies Pe > 60%

The probability that the transmission from node A to node D is successful is:

$$Pc = (1.0 - 0.1) \times (1.0 - 0.2) \times (1.0 - 0.4) = 0.432$$

Assuming a perfect back channel and that automatic repeat requests occur only at message destination points (node D in this example), it takes an average of 236 message blocks to deliver 100 correct message blocks through a channel that has 0.568 probability of transmission error. Naturally, with an imperfect back channel, this number would be higher.

1.2.4 Asymmetric Connectivity

For distributed digital radio networks, the connectivity of a node-to-node connection cannot be assumed to be symmetric. There are two different cases that illustrate this point:

Case 1:

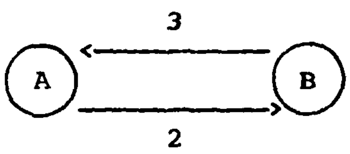


Figure 1-4. Undirected Link

For the network in Fig. 1-4, link A-B is an undirected link, but the connectivity quality CQ(A,B) is 2 and the connectivity quality CQ(B,A) is 3. The link, then, is an undirected link with asymmetric connectivity qualities. The asymmetry can be due to unequal transmission powers, different antenna heights and/or directivity, and unequal proximities of nodes A and B to a wide variety of interferers.

Case 2:

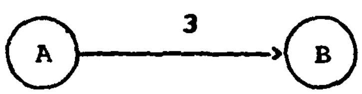


Figure 1-5. Directed Link

For the network in Fig. 1-5, the connectivity quality CQ(A,B) is 3, but

CQ(B,A) is absent in the (direct) communicant status messages. Therefore, we can say that the link is a directed link which allows communications only from node A to node B.

2.0 ISSUES IN DISTRIBUTED ROUTING

A variety of network control schemes (centralized, hierarchical, and fully distributed) can be used in a digital broadcast network, but for enhanced survivability under battlefield conditions a distributed network control scheme is considered.

Some distributed routing schemes have been proven very successful in government and commercial applications (ARPANET, slotted ALOHA, etc.), but because of various constraints associated with a digital broadcast network many computer oriented distributed routing algorithms are not applicable for use in battlefield data distribution systems. The algorithms that are being studied for applicability to the network management of a battlefield data distribution system are the Merlin-Segall protocol, Bramble-Mayk algorithm, and repromulgation relay routing.

This chapter investigates the various issues in distributed routing. The three key issues which we will investigate are: failsafe routing, minimization of control messages, and improving the network response time. We will explore these issues by citing the key properties of the various algorithms last mentioned.

2.1 Failsafe Routing

The MITRE report, "A Survey of Routing Algorithms for Distributed Digital Radio Networks" [2], states that:

1) "A common type of algorithm uses a distance matrix at each node...."

2) "At times specified by the particular scheme used, a node sends to its neighbors its minimum distance table, in whole or in part. The neighbor uses this minimum distance information to update its own distance matrix."

Thus, the distance exchange additive procedure described above requires a node to create:

1) a distance matrix;
2) a minimum distance table;
3) a routing table.

A typical distributed routing algorithm would require that each node acquire a global knowledge of the distances by iterating the distance exchange additive procedure until a net steady state is reached.

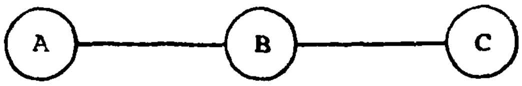


Figure 2-1. Linear Network of 3 Nodes

A distributed routing failure can be illustrated with the use of the distance exchange additive procedure. Consider the network in Fig. 2-1. Node A needs to route a message to node C. Node B informs node A that node C is one hop away from node B. Node A determines that it takes two hops to route the message to node C, and that the first hop is to node B. Node A sends the message to its preferred neighbor, node B. At the instant node B receives the message, the link B-C fails. Node B broadcasts the infinity distance to node C. Node A receives the broadcast, but the message is discarded because its present minimum distance table entry is less than infinity. Node B is informed that node A can achieve the routing in two hops. Node B enters the distance to node C as three hops and sends the message to node A. From this point on node A believes the 2 hops distance to node C, and node B believes the 3 hops distance to node C, and the message loops between node A and node B.

There *exists a routing failure (looping) if:*

1) The number of times that a message has hopped equals or exceeds the total number of nodes present.

2) The distance to the sink does not decrease after a hop.

One way to avoid looping is to implement the concept of a directed spanning tree in the routing algorithm. Merlin and Segall [3] produced a failsafe protocol which uses the directed spanning tree concept. The algorithm can be employed in packet message as well as circuit switching networks. It uses distributed computation, provides routing tables that are loop-free for all destinations at all times, adapts to changes in network flows, and is completely failsafe. Because a complete description of the protocol is rather involved the reader is referred to the original publication [3] or to the detailed application study done by MITRE Corp. [6].

Another distributed routing algorithm that is failsafe (reliable) is the repromulgation relay algorithm. The algorithm does not really route messages because it uses the method of flooding (cascading). Each message is distributed throughout the network, and if there is a connectivity between the message destination node and the rest of the network, the message is guaranteed to be delivered.

A study of the use of repromulgation relay for JTIDS was conducted by the MITRE Corp. [11]. The report used the baseline conditions used in the simulations by the Hughes Aircraft Co. [12] for PJH performance evaluation.

The report projected that 152 TS/S (timeslots per second) in timeslot capacity are needed to satisfy the total throughput requirement. This exceeds the 128 TS/S capacity available on one net.

An obvious solution to the problem is to use more than one net to satisfy the total capacity requirement. Two schemes were proposed: divide the divisional JTIDS participants into three functional repromulgation communities, or into five geographic communities. The functional communities scheme has a serious shortcoming in that "local connectivity may cause a terminal or group of terminals to be isolated from its community even though the overall connectivity for the area would seem to allow for full local connectivity" [11]. This leaves the geographic grouping scheme for consideration.

In the geographic grouping scheme, the JTIDS communities are carefully defined so that most communications are confined to within each community. Intercommunity "gateway" allocations do exist but are minimized for better timeslot resource utilization.

The repromulgation relay algorithm is reliable, but as reported, it necessitates the use of several nets to satisfy the total capacity requirement for a division-sized JTIDS community. What would be ideal is to have a reliable routing algorithm that would allow for the possibility of the use of a single net to satisfy the total capacity requirement.

An investigation of an optimal approach for timeslot allocation was conducted [10]. The paper indicates that there is the possibility of a concurrent use of timeslots by the transmitters in the same net. The repromulgation relay algorithm does not allow for the use of the optimal approach for timeslot allocation.

2.2 Message Volume

One of the important issues in distributed routing is minimizing the control messages generated for routing. A proliferation of control messages not only depletes the network capacity, but also creates unnecessary message traffic congestion which may lead to a network failure.

Capitalizing on the hierarchical nature of tactical networks which tend to constrain needlines to local nets within a global network, Bramble and Mayk [4] took a unique approach in developing a distributed routing algorithm. A limited volume of control messages is generated locally to communicate the connectivity status information necessary to satisfy the needlines associated with each pair of message origination and destination points.

The algorithm works as follows: A message origin point (source), in its attempt to locate its message destination point (sink), searches through its terminal connectivity matrix (TCM) for the sink. Typically, a TCM only maintains the connectivity quality (CQ) entries of the communicants that are within two hops of the node holding the TCM. If the source does not find its sink, it sends its direct communicants a level Ø connectivity interrogation message. Upon receiving the interrogation message, each direct communicant searches through its TCM for the sink. Of course, if the sink is further than three hops from the source no direct communicant will find the sink in its TCM.

When the source does not receive a connectivity status message within a given period of time, it generates a level 1 connectivity interrogation message in an attempt to broaden the search radius to 4 hops. The direct communicants of the source receive the level 1 interrogation message, and each takes on the function of level Ø interrogators. The level Ø interrogations are then received by the direct communicants of the level Ø interrogators, but are processed only by those who are not already participating in the search. Therefore, implicit in the algorithm is the use of the minimum spanning tree, with a flag of one kind or another indicating the nodes belonging to a minimum spanning tree. If the sink is located many hops away from the source then sequenced connectivity interrogation messages of increasing levels are broadcast by the source until the first connectivity status message is received.

The upper bound on the capacity consumed by the interrogation process in satisfying a needline is

$$TS(r) = 2[1+2+3+...+(r+1)],$$

where r is the number of relays in the route. Using the identity,

$$[\text{the sum of } i \text{ from } i=1 \text{ to } i=m] = m(m+1)/2,$$

we can restate the above relationship as

$$TS(r) = (r+1)(r+2),$$

where we've substituted r+1 for m.

When alternate routes are also considered, the capacity consumed by the interrogation process is less than or equal to s(r+1)(r+2), where s is the number of alternate routes required.

Given n=5Ø, s=2, E[r]=6, and E[NL/n]=5, where n is the number of nodes and NL/n is the number of needlines originating from a node, 28,ØØØ TS are consumed by the interrogation process. Since there are 128 TS per second, we find that it takes 3.7 minutes or less to initialize the network.

To correctly estimate the initialization time for a network, the parameters of the network must be correctly assessed. Mcquillan, Richer, and Rosen observed in their ARPANET study [5] that "although there are sites on the ARPANET separated by as many as 11 hops, about one-third of the messages in the network travel no more than one hop; about half travel no more than three hops". Furthermore, a MITRE working paper [6] projects the average number of hops per needline for a PJH network to be

E[hops/NL] given 40 nodes/network = 2
E[hops/NL] given 250 nodes/network = 3.

For most networks it is clear that the number of hops expected to satisfy a needline is rather small. Under this assumption the Bramble-Mayk algorithm should perform well. For the algorithm, global information sharing is not required, and the number of connectivity interrogations is minimized by the reception of regular broadcasts of (direct) communicant status messages.

2.3 Network Response Time

The response time of a network is defined as the amount of time required for network information such as topological changes, new needline requirements, routing, and the like to propagate through the network and reach a new steady state.

Some routing algorithms propagate update messages throughout the network when a topological change occurs. When an update message is propagated, transient loops may develop. An example of a routing algorithm that is susceptible to prolonged transient loops is the original routing algorithm for the ARPANET. The algorithm uses the distance exchange additive procedure where distance information is not only shared node-to-node but is processed by the receiving node before it is ready to broadcast the message. Although the algorithm has safeguards against permanent loops, transient loops were observed to exist for prolonged periods. The new routing algorithm for the ARPANET [5] floods the network with update messages which travel unchanged to all nodes in the network. Information is shared globally and rapidly, minimizing the durations of the transient loops.

The average length of an update message is 176 bits for the new routing algorithm compared to 1200 bits for the original routing algorithm. The update rate is less than two updates per second per link for the new routing algorithm compared to the rates for the original routing algorithm which can be as high as seven updates per second per link. It takes on the order of 100ms for all nodes to respond to a topological change using the new routing algorithm.

Mcquillan, Richer, and Rosen [5] attribute the shorter response time to using local computation instead of the global computation used by the original routing algorithm for the ARPANET. Global computation at any instant depends more on the history of events around the network than on the traffic in the network at that time. But for the local computation, because information distribution and information processing are independent processes, the traffic in the network is more accurately conveyed to all nodes with minimum delay. For these reasons local computation is better than distributed global computation.

3.0 BASIC ROUTING PROTOCOL

The basic routing protocol (BRP) works as follows: A minimum spanning tree is grown from a message origination point (source) until the message destination point (sink) is found. The path between the source and the sink found in the minimum spanning tree is the shortest path between the two points.

The protocol controls message delays by routing messages through the nodes that have the smallest operating capacities. Congestion control and minimum hop path routing are obtained by using hop distance (the number of relays required + 1) as the primary cost and path capacity as the secondary cost where the path capacity is the sum of the capacities of the nodes used in routing.

A graphics simulation program was developed. The program is used to generate various node deployment configurations. Each configuration of nodes has a network connectivity which is randomly generated. User determined needlines are established and distributed routing of messages is graphically demonstrated.

3.1 Link Cost

Mcquillan, Richer, and Rosen [5] used the packet delay as the link cost for the new routing algorithm for the ARPANET. However, they report that "The new algorithm tends to route traffic on minimum hop paths, ... ". We can use their conclusion to use the hop as the link cost instead of using the actual delay measurements. But the minimum-hop path routing isn't always an optimum routing because it doesn't take into account instantaneous or local variations in traffic conditions.

The BRP uses the capacity requirements of the nodes to determine and react to the traffic conditions. The capacity requirement of a node can be found from the original transmit (OT) timeslot requirements. (The expected number of timeslots required during an epoch/cycle for a line-of-sight needline transmission is the OT timeslot requirement.)

A node can serve as a message origination, destination, or relay point, or any combination of them. If a node serves only as a message origination or destination point its capacity requirement is the OT timeslot requirement. The capacity requirement of a relay is twice the OT timeslot requirement. Similarly, the capacity requirement of a node serving as a combination of message origination, destination, and relay point can be computed.

An algorithm can be implemented which not only seeks to find a minimum-hop path routing, but also to balance the capacity load among the nodes in the

network. BRP uses the minimum-hop distance determination first and the minimum-capacity distance determination second with each node encountered during the routing.

Consider Fig. 3-1. The network connectivity is illustrated in Fig. 3-1a. The capacity requirement associated with each link is as illustrated in Fig. 3-1b. Three possible paths exist between node A and node B as illustrated in Fig. 3-1c.

Route	Hop Distance
1	3
2	3
3	4

Table 3-1. Hops Required for Each Route

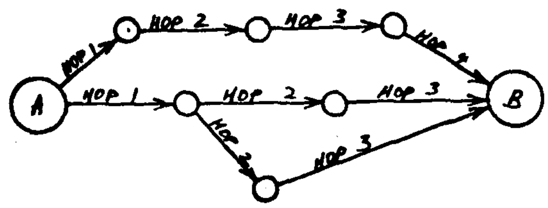
Referring to Table 3-1 route 3 is rejected because it does not satisfy the minimum-hop-path requirement.

Route No.	Link 1	Link 2	Link 3	TOTAL
1	1	1	1	3
2	1	2	1	4

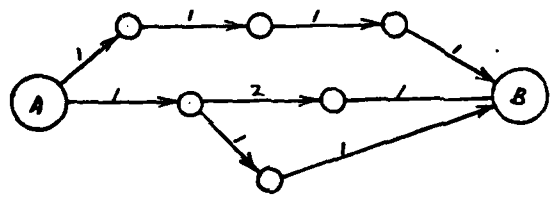
Table 3-2. Capacity Distance Table

Although the hop distances for route 1 and route 2 are equal, the capacity distances are different, as shown in Table 3-2. Route 1 would be the logical routing choice to satisfy the node A - node B needline. (Note that the link weights represent capacity requirements (costs) and not capacity availability.)

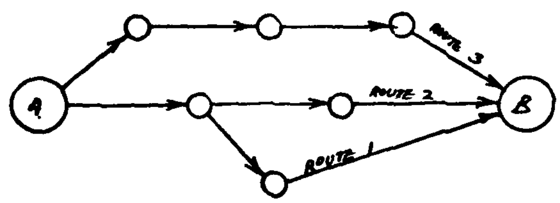
The next section investigates two problems in connection with graphs: the minimum total branch cost problem and the minimum path cost problem. The second problem is of particular interest because the problem deals with finding the path of minimum total length between two given nodes.



a. Network Connectivity



b. Link Capacities



c. Available Paths

Figure 3-1. A Simple Routing Problem

3.2 Graph Theory and Routing

In this section, a discussion of graph theory precedes the discussion of the minimum total branch cost problem and the minimum path cost problem. For a more detailed discussion of graph theory, the reader is referred to Gallager [13].

A graph is represented by G(V,E), where V is a set of vertices (nodes) and E is a set of edges (links).

A tree is a graph with one and only one path between every two nodes. A graph which has loops, or a graph which is cyclic, is not a tree. A tree is defined as an acyclic connected graph.

A directed spanning tree is used to find a route between two given nodes. A directed spanning tree has a root which initiates the growth of the tree. (The BRP assigns the message source as the root of the tree). Starting from the root, the tree grows until the sink is found in the tree. A path between the source and the sink is effectively established if the source and the sink coexist within the tree.

E. W. Dijkstra [7] produced a note on two problems in connection with graphs which is of interest to our present discussion. We've explored the two problems with the use of a computer graphics simulation. The results obtained are discussed in the next two sections.

3.2.1 Minimum Total Branch Cost Problem

Consider a situation in which there are n nodes, each linked to at least one other node such that all nodes are connected, either directly or indirectly. Initially, all nodes are in the set

OUTSIDE = [node (1), node (2), ..., node (n)],

and there is no node in the TREE. The first given node n(i) is transferred from the set OUTSIDE to the set TREE such that

TREE = [n(i)];

n(i) is now the root of the tree. The rest of the nodes from the set OUTSIDE are transferred to the set TREE in sequence and in such a manner that the node from the set OUTSIDE that has the shortest link to any of the nodes in the set TREE is transferred first. With each transfer of a node, the link that was used as the shortest link becomes a branch of the tree. If all nodes are connected, either directly or indirectly, the set TREE will consist of n nodes, and the set OUTSIDE will be an empty set.

This algorithm is designed to generate a tree of minimum total length among n given nodes. This algorithm is not of use to us because it does not produce a path of minimum total length between the root of the tree and any other given node. The algorithm, however, is applicable to problems like finding the procedures for connecting terminals with a minimum total cost.

A computer graphics simulation program was used to demonstrate the algorithm at work. When compared to the simulation result found in Fig. 3-3, the simulation result obtained in Fig. 3-2 is clearly a tree of shorter total length connecting the given 30 nodes.

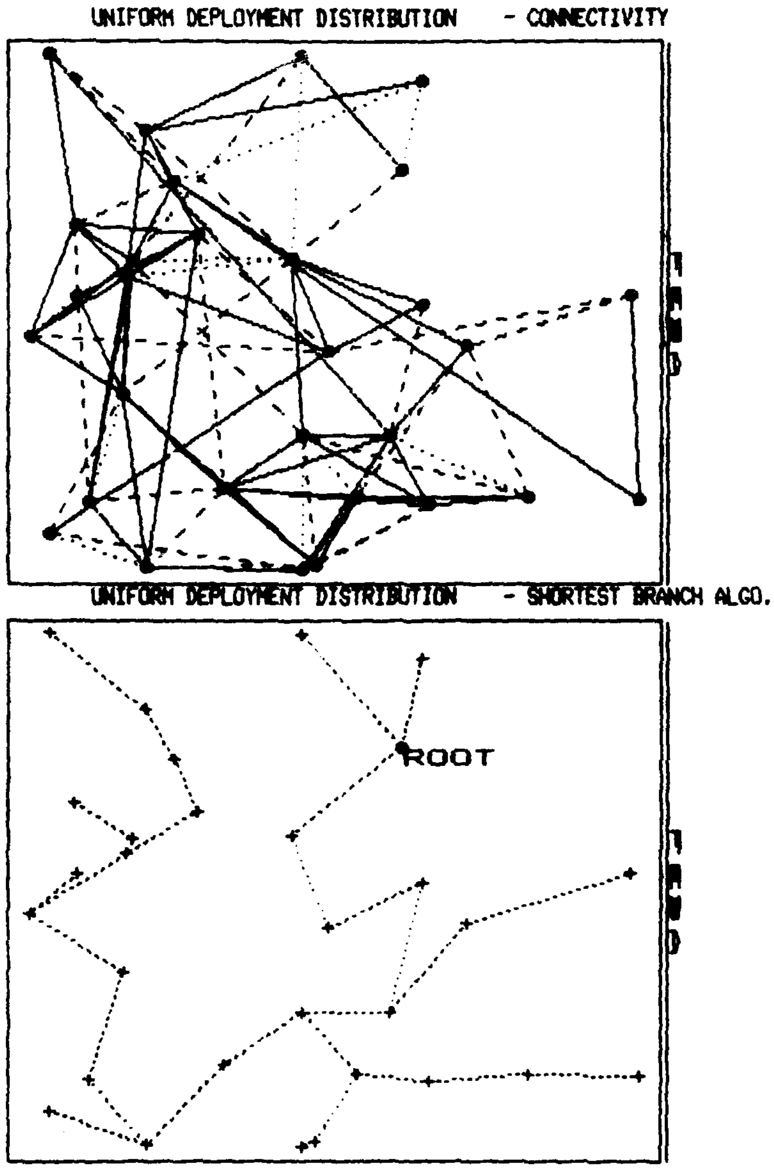


Figure 3-2. Minimum Total Branch Cost Algorithm Graphics Output

3.2.2 Minimum Path Cost Problem

Consider the same situation described in the minimum total branch cost problem. Initially, all nodes are in the set

OUTSIDE = [node (1), node (2),..., node (n)],

and there is no node in the set TREE. The first given node n(i) is transferred from the set OUTSIDE to the set TREE such that

TREE = [n(i)];

n(i) is now the root of the tree. The rest of the nodes from the set OUTSIDE are transferred to the set TREE in sequence and in such a manner that the node from the set OUTSIDE that has the shortest distance along the existing links to the root of the tree is transferred first. (Distance is defined as: link cost (source, relay 1) + link cost (relay 1, relay 2) + ... + link cost (relay m, sink).) With each transfer of a node, the link that was used for shortest distance and which wasn't a branch of the tree becomes a branch of the tree. If all nodes are connected, either directly or indirectly, the set TREE will consist of n nodes, and the set OUTSIDE will be an empty set.

This algorithm is used to find the path of minimum total length between two given nodes. For this algorithm to produce the desired path, however, either of the two given nodes must be the root of the spanning tree.

The graphics simulation output in Fig. 3-3 shows a spanning tree generated using the shortest path algorithm. The total length between the root and any other node is the minimum total length between the two nodes. Note that although the minimum total length between the root and any other node is achieved with the use of the shortest path algorithm, the total length of all the links is greater for the tree in Fig. 3-3 when compared to the total length of all the links for the tree in Fig. 3-2.

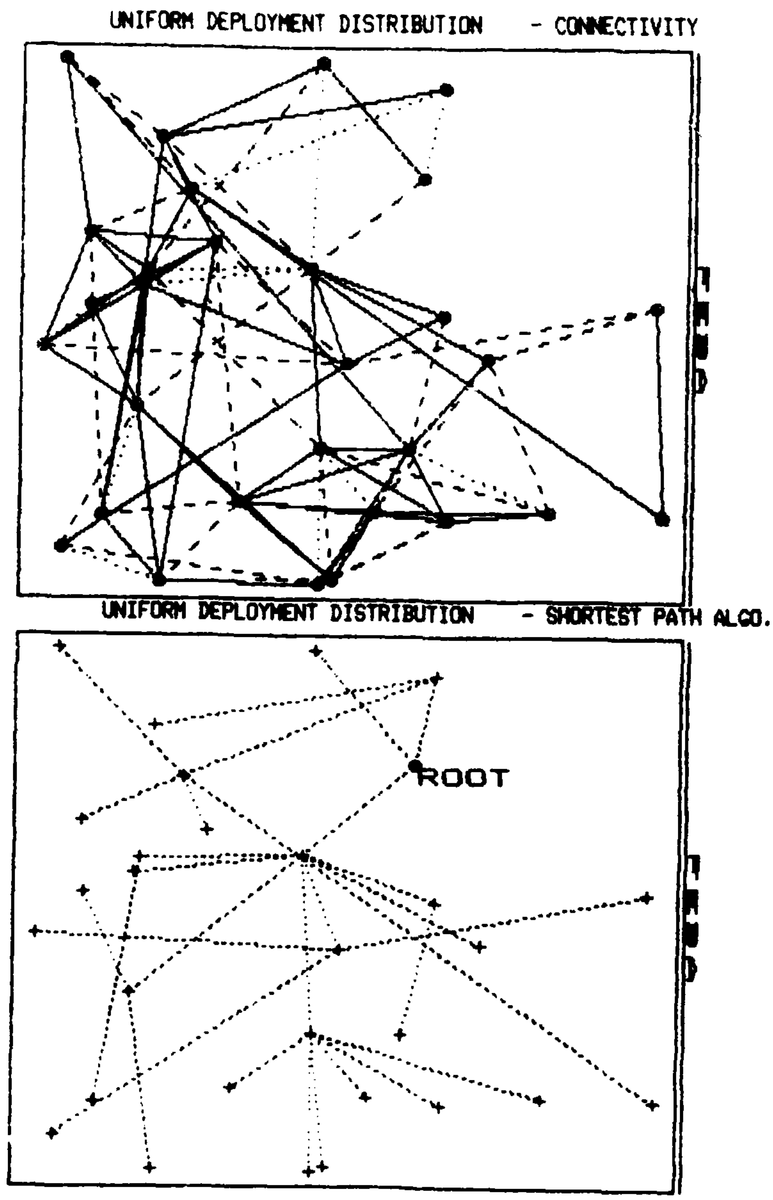


Figure 3-3. Minimum Path Cost Algorithm
Graphics Output

3.3 Hop Levels

When a path cost is defined in terms of hops, each link cost carries a value of unity. The distance between any two given points is the number of hops required to route a message from the origination point to the destination point.

The hop level of a node indicates the number of hops required for a message from the source to reach the node. A minimum spanning tree grows in such a manner that the nodes belonging to the next hop level become a part of the minimum spanning tree. Consider the following scenarios.

Scenario No. 1. Level 1 Propagation

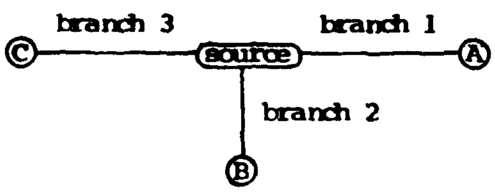


Figure 3-4. Level 1 Propagation

Referring to Fig. 3-4, nodes A, B, and C belong to level 1. They are branched to in node sequence of A, B, and C, because the node capacities are

Capacity cost (node A) ≈ 1
Capacity cost (node B) ≈ 2
Capacity cost (node C) ≈ 3
Capacity cost (source) ≈ 1

Scenario No. 2. Level 2 Propagation

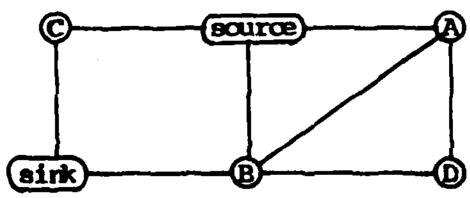


Figure 3-5. Level 2 Propagation

Referrring to Fig. 3-5, the nodes that belong to level 2 are node D and the sink. The node capacities are

Capacity cost (node D) = 1
Capacity cost (sink) = 1

The source, having completed the level 1 phase, requests nodes A, B, and C to submit a branch of opportunity. Node A has 3 links to choose from:

CQ(A,source)
CQ(A,B)
CQ(A,D).

The links associated with CQ(A, source) and CQ(A,B) are rejected because the source and node B belong to the set TREE = [source, A, B, C] indicated by a flag. Node D associated with the remaining link CQ(A,D) is chosen as the preferred neighbor of opportunity and the following assignments are made:

Hop distance (source, D) ---> SHORT (node A)
Capacity distance (source, D) ---> LESS (node A)
node D ---> BETTER (node A)

Node A submits the three variable assignments to the source.

Next consider the equivalent processing at node B. Node B detects the presence of the sink, requests the sink to submit data, and makes the following assignments.

Hop distance (source, sink) ---> SHORT (node B)
Capacity distance (source, sink) ---> LESS (node B)
sink ---> BETTER (node B)

Node B submits the three variable assignments to the source.

Finally, consider the equivalent processing undertaken at node C. Node C detects the presence of the sink, requests the sink to submit data, and makes the following assignments:

Hop distance (source, sink) ---> SHORT (node C)
Capacity distance (source, sink) ---> LESS (node C)
sink ---> BETTER (node C)

The source, therefore, receives the information submitted by all peripheral nodes of the minimum spanning tree (PNMST) and finds that a sink detection has occurred. The information submitted by node A is discarded because the information does not pertain to the detected sink. However, the source has two paths to the sink to choose from, both of which are of the same hop

distance. An obvious choice would be the path that has the least expected congestion. Table 3-3 illustrates the capacities that are involved.

	CAPACITY COSTS			
Route	Source	Relay	Sink	TOTAL
Source-B-Sink	1	2	1	4
Source-C-Sink	1	3	1	5

Table 3-3. Capacity Distance Table

The source chooses the Source-B-Sink path because the capacity cost of the Source-B-Sink path is less than the capacity cost of the Source-C-Sink path.

Fig. 3-6 illustrates the decision process occurring at the root of a minimum spanning tree. Fig. 3-7 illustrates the decision process occurring at a peripheral node of a minimum spanning tree. The two processes are coordinated for the growth of a minimum spanning tree.

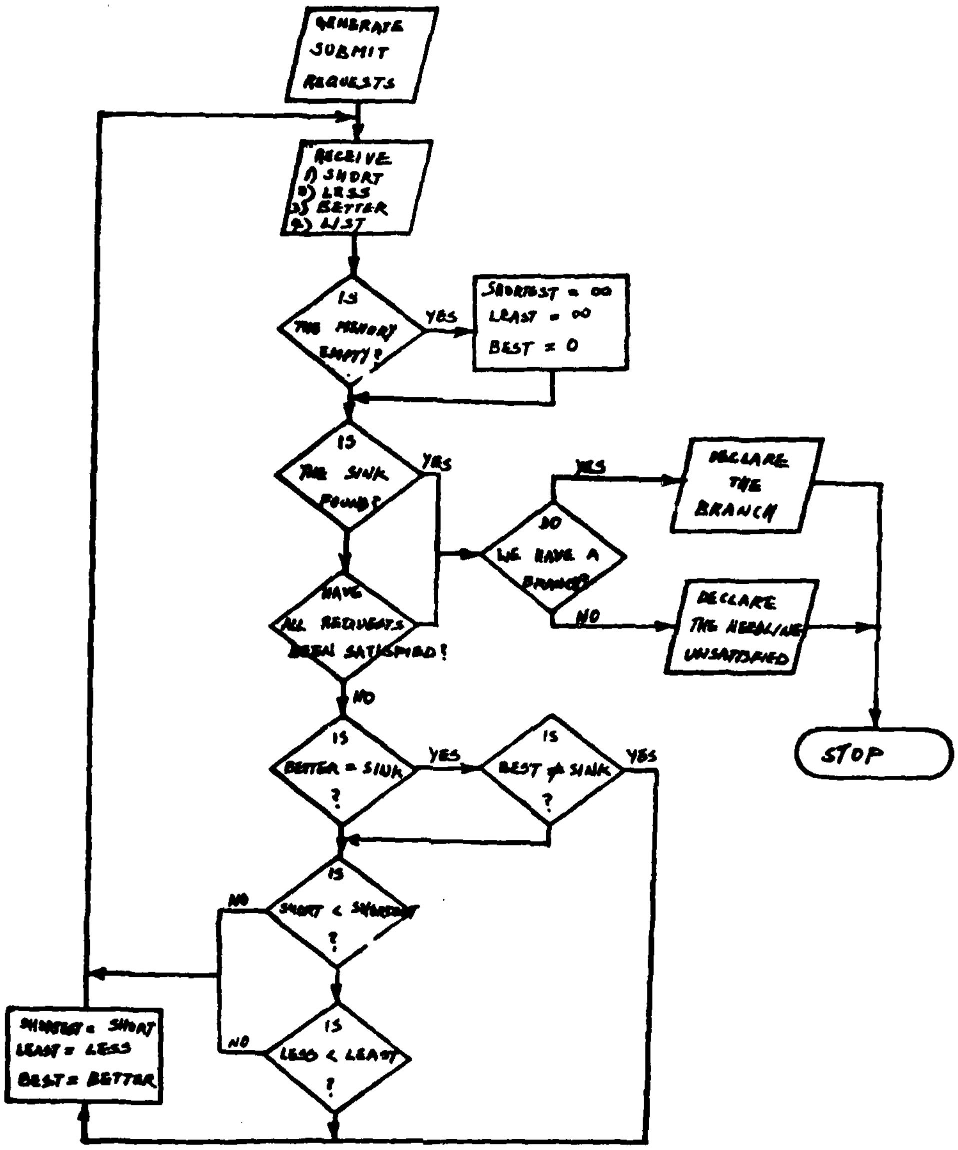


Figure 3-6. Decision Process at the Root

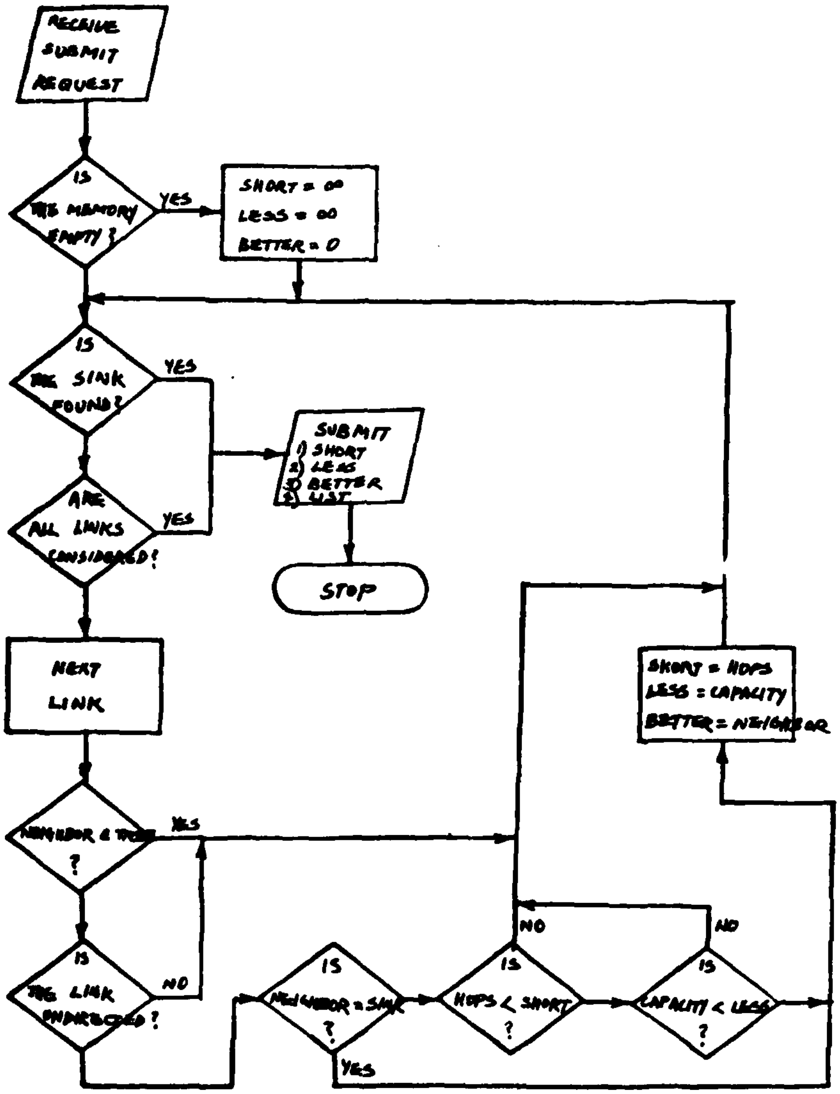


Figure 3-7. Decision Process at a Peripheral Node

3.4 Routing List

A routing list is a sequenced list of the nodes required to construct a route from the source to the destination, starting with the source terminal I.D. No. and ending with the terminal I.D. No. of the destination.

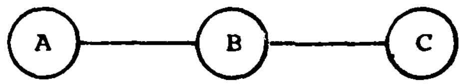


Figure 3-8. A 3-node Linear Network

A minimum spanning tree with the source as its root searches for the sink through a growth process. At any time prior to sink detection it is essential for the source to keep a complete knowlege of the routing requirements of all PNMSTs. Consider the network in Fig. 3-8.
Node A is the source and node C is the PNMST. The source may keep the following routing lists:

List (A) = [A]
List (B) = [A,B]
List (C) = [A,B,C]

To reduce the memory requirements of the source, however, the source keeps only the routing list of the PNMST. Consider the network in Fig. 3-9.

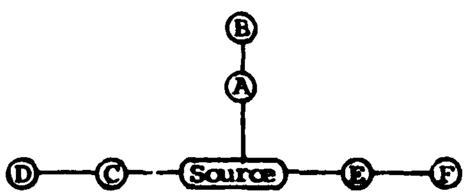


Figure 3-9. A 7-node Network

The source keeps the following routing lists:

List (B) = [Source,A,B]
List (D) = [Source,C,D]
List (F) = [Source,E,F]

The source does not keep list (C), list (A), and list (E).

3.5 Routing Failure

A loss of a route can occur when a loss of connectivity is experienced. A routing failure can be avoided if a loss of route is detected and corrected by having the source reinitiate the routing procedure. The key to detection is the routing list.

Figure 3-10. 4-node Linear Network

Consider the network in Fig. 3-10. Node A has a route to node D as follows: A - B - C - D. Node A sends its message with the routing list

List (D) = [A B,C,D].

Node B receives the message and sends it to C. Node C receives the message, determines that the next node in sequence is node D, but finds that the link CQ(C,D) no longer exists. It returns a new routing list of

List (A) = [C,B,A]

which has the reverse sequence of [A,B,C]. When node A receives the message it reinitiates its routing procedure to find an alternate path to node D.

3.6 Multiple Needlines from a Common Source

It was previously stated that a routing procedure is initiated upon realizing a needline. This implies that a separate minimum spanning tree is initiated for each needline. However, a reduction in the time it takes to initialize the network can be achieved when a common tree is used for needlines from a common source. Under this scheme, the minimum spanning tree propagation is halted when the number of sinks that are located is equal to the number of needlines originating from the common source.

3.7 Simultaneous Occurrence of Minimum Spanning Trees

As described previously, the variables used by the BRP are not capable of handling a simultaneous occurrence of minimum spanning trees. There are two solutions to the problem, each with its advantages.

Solution No. 1:

All variables will be modified to accommodate a new array argument that identifies the minimum spanning tree that each variable pertains to. This method achieves the capability for handling simultaneous occurrence of minimum spanning trees at a cost of an increased memory requirement of the terminal processors.

Solution No. 2:

No modification of the existing variables is required but the occurrence of the minimum spanning trees is sequentially orchestrated so that no two minimum spanning trees intersect a node at any given time. This method achieves a substantial reduction of the memory requirement of the processors at a cost of taking longer to initialize the network.

3.8 BRP Simulation Results

A graphics simulation program was developed to observe the state of a network as a function of time. The program is capable of deploying up to 300 nodes using any of the four random distributions:

1. uniform
2. gaussian
3. Poisson
4. exponential

Network connectivity matrix can be randomly generated: each node-to-node connectivity quality (CQ) is described by a 4-ary quality field generated by an exponential distribution random number generator. The user controls the average degree per node so that a wide range of network connectivities can be generated.

Although there can exist any number of needlines in a network, as long as the number does not exceed n(n-1), where n is the number of nodes in the network, for the purpose of obtaining a clear visual understanding of the graphics demonstrations we use a small number of needlines.

Fig. 3-11 shows the connectivity of a network of 30 randomly deployed nodes. Figs. 3-12 and 3-13 show the needlines used and the routes established to satisfy the needlines. We find that all paths are minimum-hop paths.

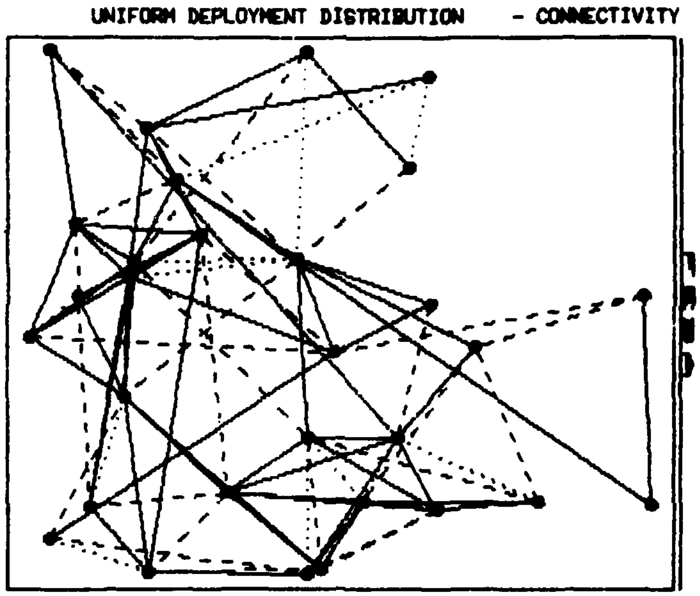


Figure 3-11. Network Connectivity

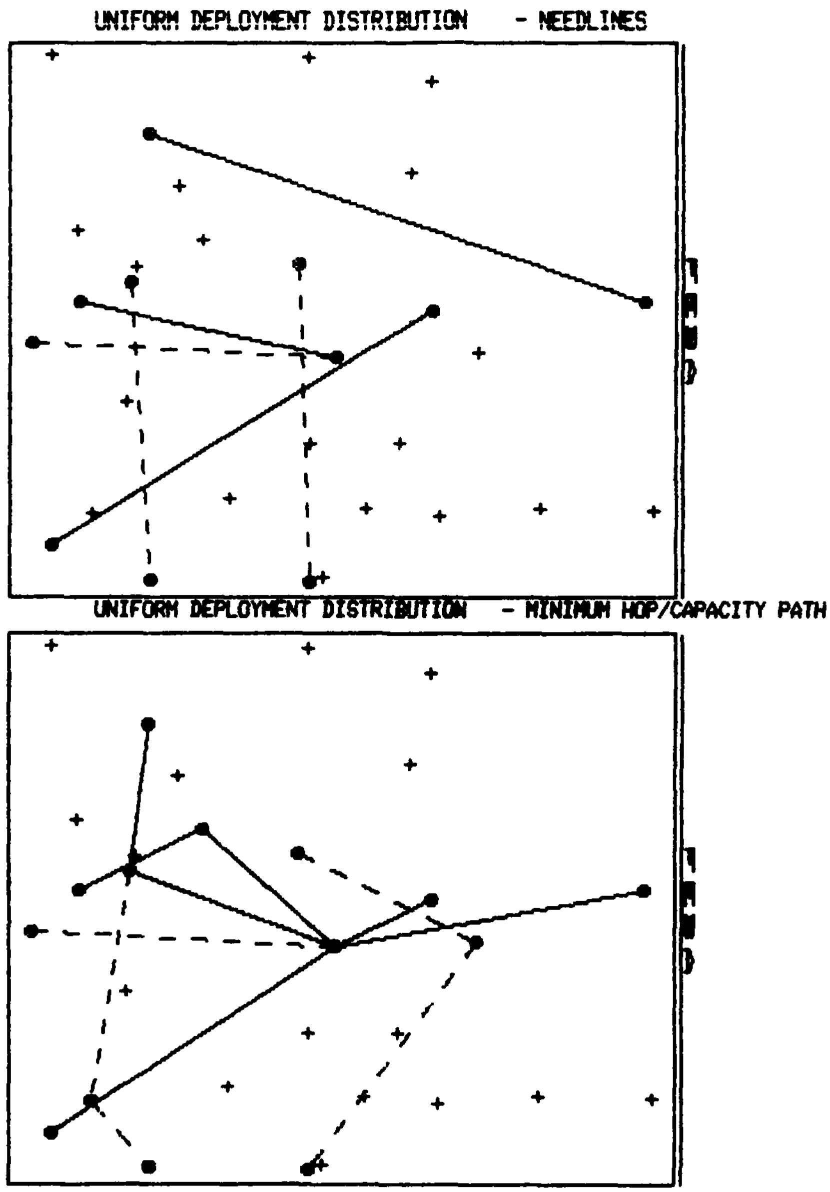


Figure 3-12. BRP Simulation Output for 6 Needlines

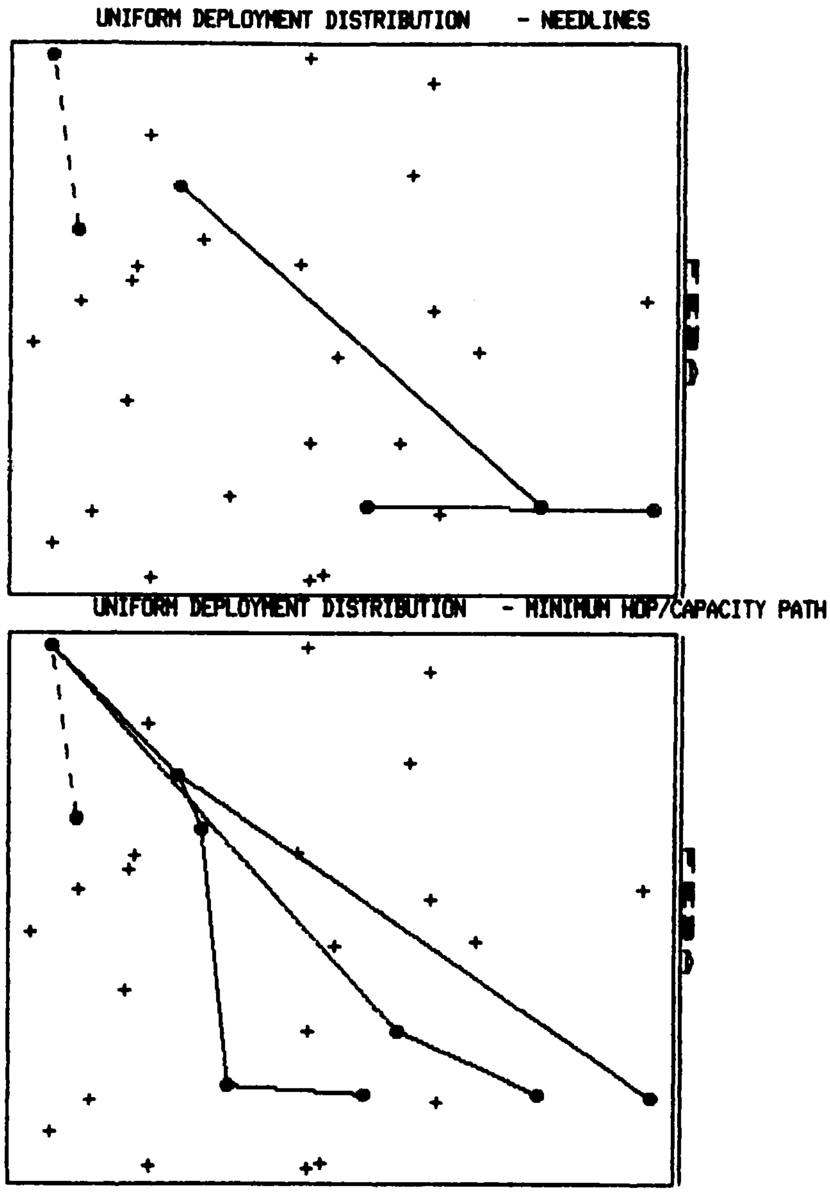


Figure 3-13. BRP Simulation Output
for 3 Needlines

4.Ø EARLY DETECTION ROUTING PROTOCOL

The early detection routing protocol (EDRP) uses the BRP and the communicant status (level Ø) messages (CSMØ) to achieve distributed routing. Although the BRP is sufficient to route messages, when CSMØs are used much of the information acquisition in distributed routing is achieved without the use of interrogations.

4.1 Communicant Status (level Ø) Message

A communicant status (level Ø) message (CSMØ) is a control message which is periodically broadcast by each terminal. It lists the connectivity qualities (CQs) of the broadcaster's direct communicants. A node that receives CSMØs not only knows the CQs of its direct communicants, but it also knows the CQs of the direct communicants of its direct communicants through the use of its terminal connectivity matrix (TCM).

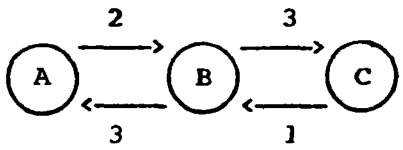


Figure 4-1. A 3-node Linear Network

Consider the network in Fig. 4-1. Nodes A, B, and C enter the net in sequence. Upon entering the net each node regularly broadcasts and receives a CSMØ. With each reception of a CSMØ each node updates its terminal connectivity matrix (TCM) and its CSMØ broadcasts. Table 4-1 summarizes the complete update events of the CSMØ broadcasts that occur.

Count	Sender	CQ(A,B)	CQ(B,A)	CQ(B,C)	CQ(C,B)
1	A				
2	B	2			
3	A	2	3		
4	B	2	3		
5	C			3	
6	B	2	3	3	1
7	C			3	1

Table 4-1. Communicant Status (level Ø) Messages

The CSMØ that a node receives is used to update the node TCM. With enough updates, the node gains a complete bidirectional CQ knowledge of all links within two hops from the node.

4.2 Terminal Connectivity Matrix

The connectivity data base of a terminal is a matrix referred to as the Terminal Connectivity Matrix (TCM) which is contained in each TDMA terminal. (To be constistent with the rest of the text, it should be pointed out that the words "terminal" and "node" are used interchangeably in this section.) The TCM is used to store information on any terminal's connectivity with other netted terminals. Since, in a non-line-of-sight environment, the connectivity of a particular terminal to other terminals in the net is incomplete, each terminal will have a TCM which is said to be local.

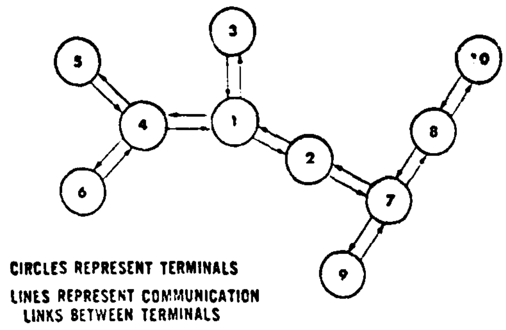
Using Fig. 4-2b as the reference we make the following observations:

1) The matrix is symmetric in the sense that if CQ(i,j) is a nonzero entry then CQ(j,i) is also a nonzero entry. The symmetry of the matrix allows us to analyze the data by looking only at the upper diagonal matrix (UDM).

2) The entries of the first row of UDM tell us that terminals 2, 3, and 4 are the direct communicants of terminal 1.

3) The entries of the second and fourth row of UDM tell us that terminals 5, 6, and 7 are the indirect communicants of terminal 1. To establish the terminal 1 - indirect communicant communications links, the following routing arrangements are necessary:

   a. 1 - 2 - 7
   b. 1 - 4 - 5
   c. 1 - 4 - 6



a. A Network of 10 Nodes

	1	2	3	4	5	6	7	8	9	10
1		CQ_{12}	CQ_{13}	CQ_{14}						
2	CQ_{21}						CQ_{27}			
3	CQ_{31}									
4	CQ_{43}				CQ_{45}	CQ_{46}				
5				CQ_{54}						
6				CQ [illegible]						

4.3 Early Detection

Early detection of the sink occurs in two phases. A PNMST will detect the sink in the following manner:

1) A PNMST first detects the presence of the sink two hops away from the minimum spanning tree.
2) The PNMST later detects the presence of the sink one hop away from the minimum spanning tree.

4.3.1 First Level Early Detection (FLED)

When FLED occurs, the PNMST that detects the sink chooses the sink as its preferred neighbor of opportunity, interrogates the sink for its node capacity, and halts its information gathering process. The PNMST submits the branch-of-opportunity data to the source.

The source, however, does not halt its information processing upon receiving the data containing the sink information, because there may be other incoming data also containing the sink information which may indicate a better route to the sink. The source gathers all branch-of-opportunity data submitted by each PNMST, rejects the branch-of-opportunity data not containing the sink information (if a sink data exists within the set of submitted data), and selects the best choice among the remaining data. The reader is referred to the BKP flowcharts for details of the decisions involved in the algorithm.

4.3.2 Second Level Early Detection (SLED)

When SLED occurs, the PNMST that detects the sink two hops away from itself chooses the direct communicant which is a direct communicant of the sink as its preferred neighbor of opportunity, but unlike in the previous case, the information gathering process continues until either FLED occurs or the links to be considered have been depleted. The PNMST requests the preferred neighbor of opportunity for the node capacity, and submits the branch-of-opportunity data to the source.

The source gathers the data submitted by each PNMST, rejects the branches of opportunity not indicated by SLED, and selects the best choice among the remaining data.

It is possible that FLED and SLED can simultaneously occur for a given submit request cycle. However, they can also occur simultaneously during the information gathering process of a PNMST. A PNMST may detect the sink

two hops away from itself, but the importance of the detection is only second to the importance of FLED, therefore the SLED algorithm exists seperate from the FLED algorithm and takes second priority to the FLED algorithm.

4.3.3 Early Detection Algorithm

Early detection can be achieved by methodically searching through the terminal connectivity matrices of each PNMST for the sink entry. For simplicity, we will assume that all links are undirected, which implies that all TCMs are symmetric in the sense that if there exist a nonzero CQ(i,j) entry, then there exists a nonzero CQ(j,i) entry regardless of the actual CQ values.

Referring to the TCM of Fig. 4-2b which is the connectivity data base of terminal 1 (found in Fig. 4-2a), we will assume the following:

1) All links are undirected.
2) Terminal 1 is the source.

If we define an entry CQ(i,j) as the connectivity quality of the link i-j in the direction going from i to j, then we can ignore the lower diagonal entries of the TCM and work only with the upper diagonal matrix (UDM) because the entries of the LDM represent the connectivities of the reverse paths.

Looking at the first row entries, terminals 2, 3, and 4 are terminal 1's direct communicants. If any of the terminals is the sink, then FLED has occurred.

Looking at the entries of the second and fourth rows, terminals 5, 6, and 7 are the indirect communicants of terminal 1. If any of the terminals is the sink, then SLED has occurred. For terminal 1, node 4, 4, or 2 is the preferred neighbor of opportunity if node 5, 6, or 7 is the sink, respectively.

4.4 EDRP Simulation Results

A graphics EDRP simulation program was developed which has the same user interaction format as the BRP simulation program. The reader is referred to the appendix for a full description of the program.

Fig. 4-3 shows the connectivity of a 17x17 Manhattan network. Fig. 4-4 shows the needlines used and the routes established to satisfy the needlines.

Fig. 4-5 shows the connectivity of a network of 300 randomly deployed nodes. Figs. 4-6 and 4-7 show the needlines used and the routes established to satisfy the needlines.

As expected, all paths shown are minimum-hop and minimum-capacity paths. This means that it is possible to maintain a uniform capacity distributed network that is capable of failsafe distributed routing.

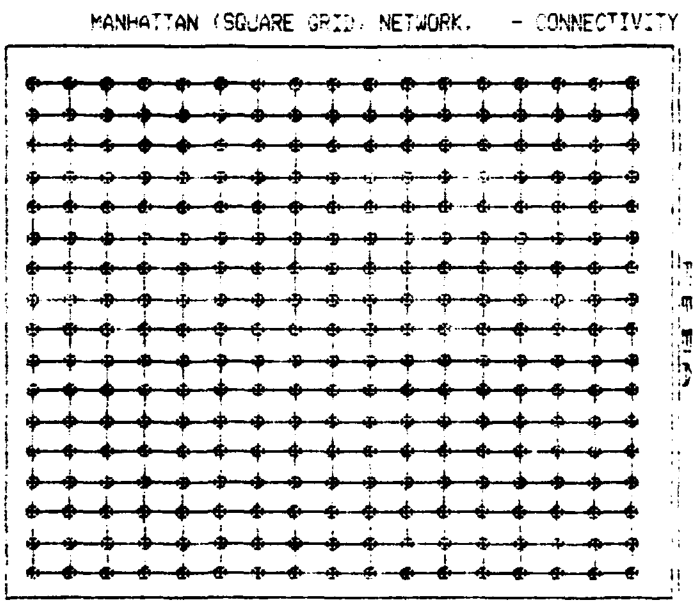


Figure 4-3. A Manhattan Network

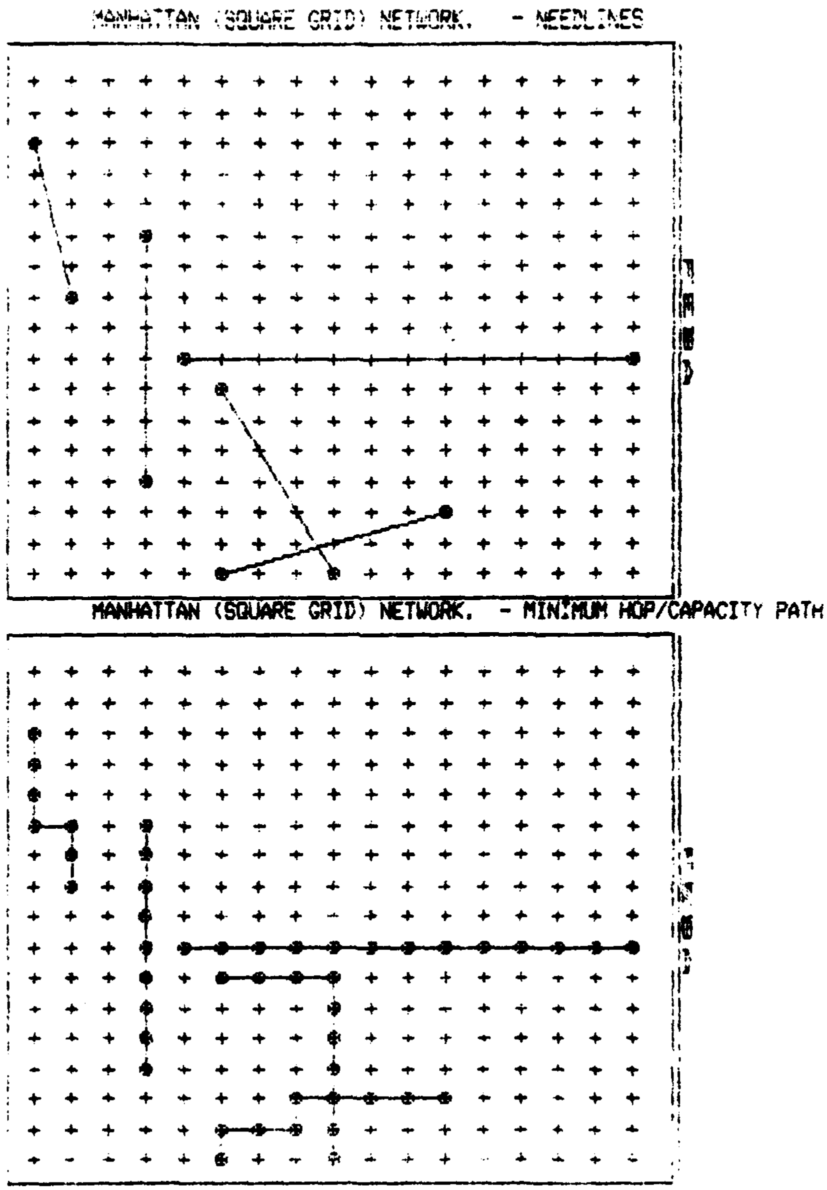


Figure 4-4. EDRP Graphics Output for a Manhattan Network

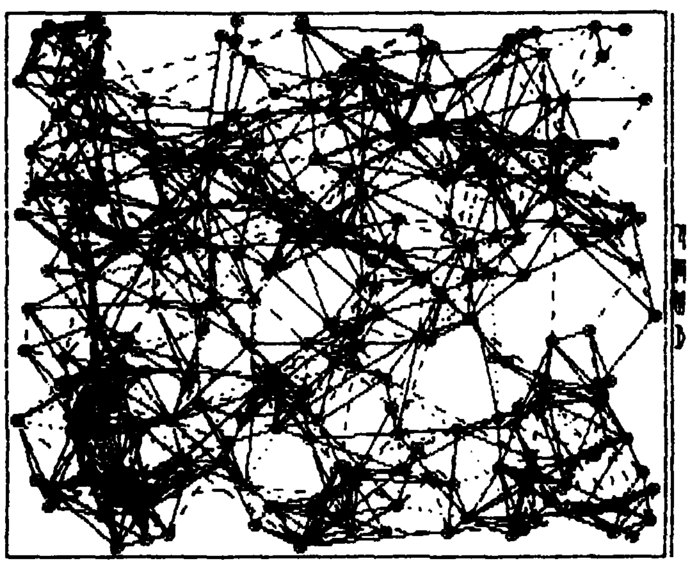


Figure 4-5. Network Connectivity

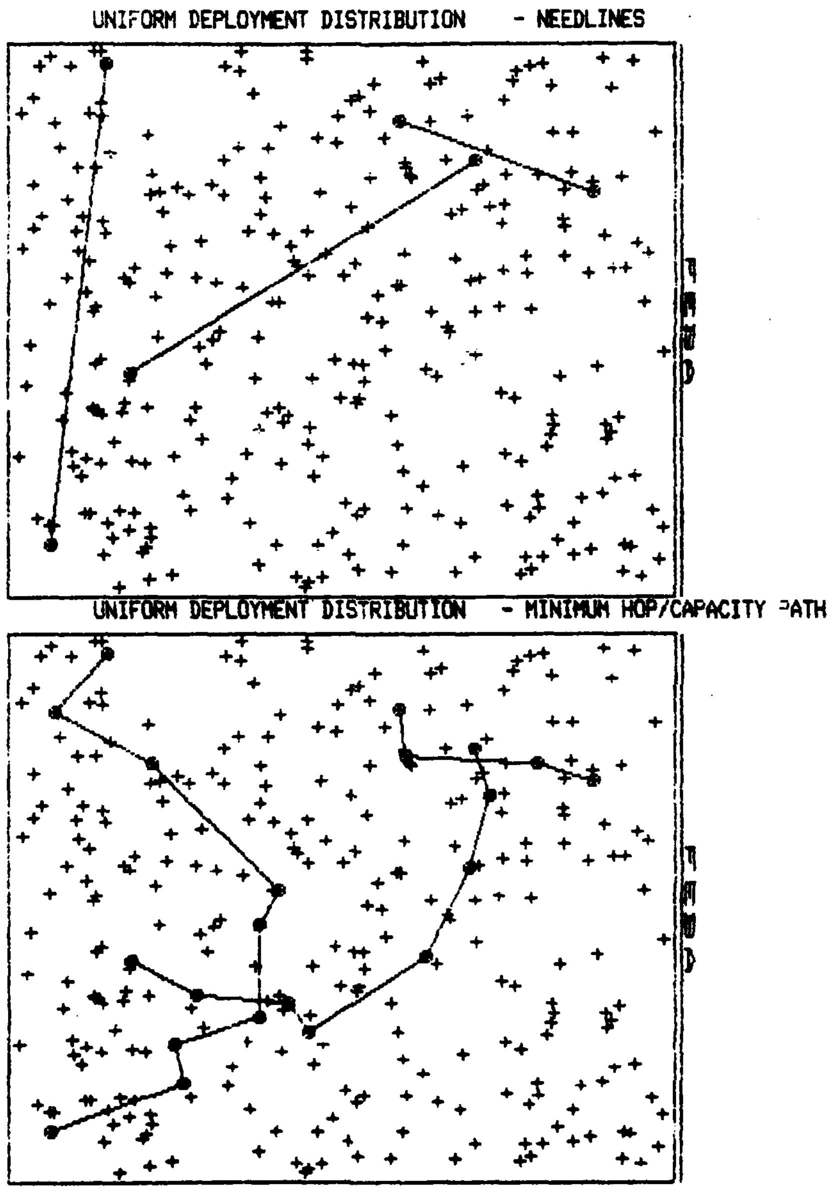


Figure 4-6. EDRP Graphics Output for a Random Deployment of Nodes

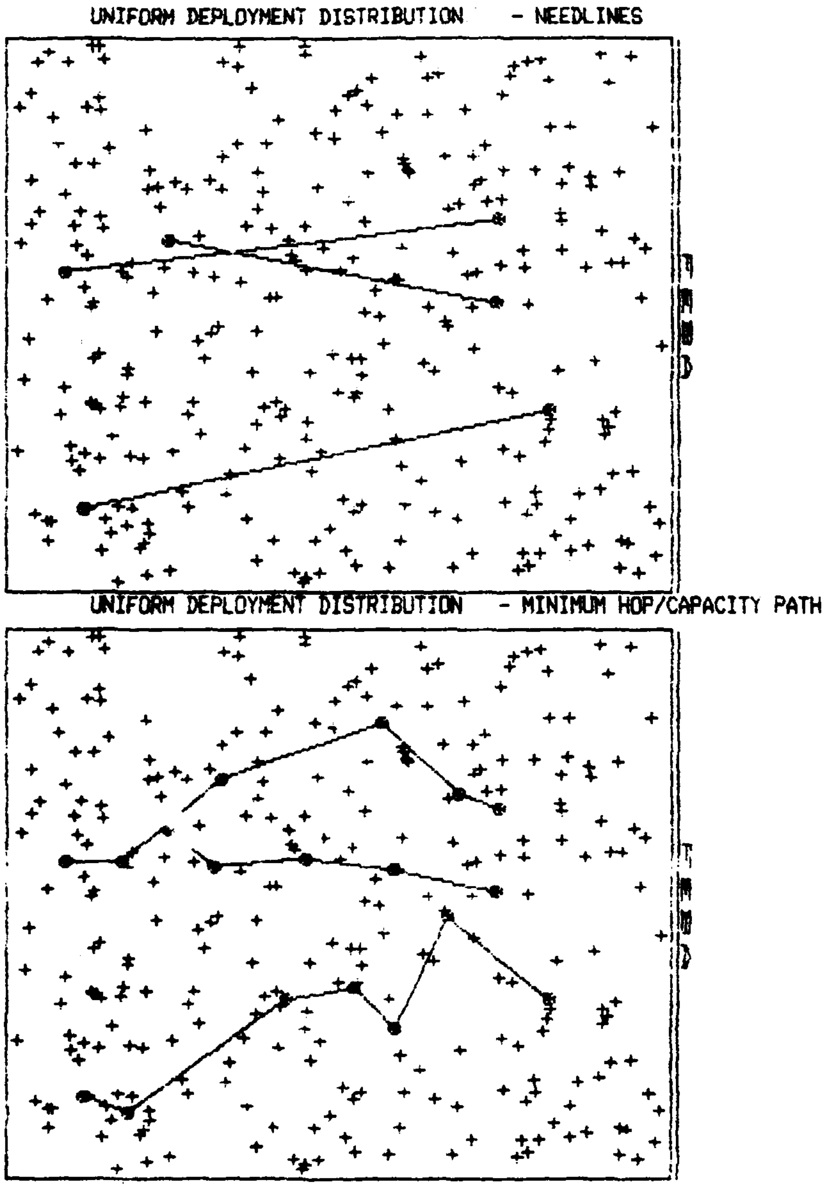


Figure 4-7. EDRP Graphics Output for a Random Deployment of Nodes

5.0 CONCLUSIONS

The graphics simulation results show that the EDRP works for many different networks. The paths that were generated to satisfy the needlines were consistently minimum-hop and minimum-capacity paths.

An exciting avenue of approach towards achieving an optimum timeslot allocation in TDMA networks exists as indicated by Skiscim [10]. This approach, unlike the commodity flow approach that we've taken, seeks to optimize the throughput by looking at the individual traffic conditions for each timeslot. As part of our ongoing research we will consider this approach, first looking at it from the theoretical point of view, and then assessing its implementability based upon the parameter constraints of a typical digital broadcast network.

Currently, the authors are involved in an ongoing effort to develop a simulation tool for the network manager of a battlefield data distribution system. The simulation tool is envisioned to be capable of optimum routing, optimum capacity distribution, adapting to topological changes, and reacting to changes in the communications needs of the net subscribers.

ACKNOWLEDGMENT

The authors would like to thank Mr. Eli Goldman, a colleague at Fort Monmouth, for his helpful suggestions during the writing of this report.

REFERENCES

1. Joint Tactical Information Distribution System (JTIDS) Planning Guide (Revision 4), The TAF Net Mangement Working Group, September 1983.

2. Carpenter, R.W. A Survey of Routing Algorithms for Distributed Digital Radio Networks, the MITRE Corp., March 1981.

3. Merlin, P.M. and Segall, A., "A Failsafe Distributed Routing Protocol", IEEE Transactions on communications, Vol. COM-27, No. 9, September 1979.

4. Bramble, R. and Mayk, I., "A New Routing Algorithm for Distributed Processing in TDMA JTIDS", US Army CECOM Research and Development Technical Report (unpublished), February 1982.

5. Mcquillan, J.M., Richer, I., and Rosen, E.C., "The New Routing Algorithm for the ARPANET", IEEE Transactions on Communications, Vol. COM-28, No. 5, May 1980.

6. PJH Decentralized Network Management: Fully Distributed Routing, The MITRE Corp., WP83W00079, April 1983.

7. Dijkstra, E.W., "A Note on Two Problems in Connexion with Graphs", Numerische Mathematik, Volume I, pp. 269-271, 1959.

9. Silvester, J. and Kleinrock, L., "On the Capacity of Multihop Slotted ALOHA Networks with Regular Structure", IEEE Transactions on Communications, Vol. COM-31, No. 8, August 1983.

10. Skiscim, C., "An Optimal Approach for Timeslot Allocation in TDMA Networks", 2nd Annual IEEE Conference on Military Communications, October 1983.

11. Jablokow, A. and Rice, W., The Use of Repromulgation Relay for JTIDS, The MITRE Corp., MTR8786, November 1982.

12. PLRS/JTIDS Hybrid Definition and Evaluation Final Report, Hughes Aircraft Co., September 198C.

13. Gallager, R.G., Data Communication Networks, unpublished.

APPENDIX

EDRP Graphics Simulation

An EDRP graphics simulation program was developed at the CENSEI Systems Simulations Facility using the ReGIS Graphics Library. The VAX FORTRAN program runs on a VAX 11/780 computer with the VT125 terminal used for user interaction.

The program is divided into several segments: deployment of the nodes, generation of the network connectivity, generation of a set of desired needlines, and routing execution. The routing scenario can be printed out on a graphics printer along with the deployment, connectivity, and needline scenarios.

In the deployment segment of the program, the user chooses a uniform, gaussian, Poisson, or exponential node deployment distribution. For a user interested in a theoretical research in network management there is available a Manhattan network option which produces an nxn grid network, where n is an integer specified by the user. The number of nodes in the network ranges from 1 to 300.

In the network connectivity segment, the program generates a random network connectivity using the exponential distribution function number generator. (The exponential distribution function has a parameter Lambda which determines the spread of the connectivity likelihood region.) The user specifies the expected degree of a node. The user can produce a connectivity rich network by specifying a large expected degree value, or a connectivity poor network (or even a disconnected network) by specifying a small expected degree value.

In the needlines segment, the program generates a random pattern of needlines using the uniform distribution function number generator. The user specifies the number of needlines to be used, but the program does not generate more than n(n-1) needlines, where n is the number of nodes deployed. It is also possible to generate needlines with the use of a graphics cursor. The user controls the cursor movement with the use of the arrow keys on the keyboard.

There is a segment that graphically demonstrates the shortest-path algorithm which produces a minimum-spanning tree rooted at a node which the user specifies. For clarity in analyzing the results, the physical distance that seperates each pair of nodes is taken as the link cost. The results of an execution of the algorithm are strongly influenced by the network connectivity that is used, so the segment can be a valuable tool for a user who is interested in investigating the effects of differing network connectivity on network management.

Finally, there is a segment which graphically demonstrates the distributed routing of messages using the EDRP. There is no input parameter that the user needs to specify. The parameters that are used are the deployment, connectivity, and needline data that were previously generated and stored in variable arrays.

The authors are in the process of upgrading the EDRP graphics simulation program. The program listing is not available for distribution at this time.

LIST OF ACRONYMS

BRP	basic routing protocol
CQ	connectivity quality
CSMØ	communicant status message (level Ø)
EDRP	early detection routing protocol
FLED	first level early detection
JTIDS	Joint Tactical Information Distribution System
LDM	lower diagonal matrix
OT	original transmit
PJH	PLRS/JTIDS Hybrid
PNMST	peripheral node of the minimum spanning tree
SLED	second level early detection
TCM	terminal connectivity matrix
TDMA	time division multiple access
TS/S	timeslots per second
UDM	upper diagonal matrix

ATE

MED